

Robust Multidimensional Mean-Payoff Games are Undecidable

Yaron Velner

The Blavatnik School of Computer Science, Tel Aviv University, Israel

Abstract. Mean-payoff games play a central role in quantitative synthesis and verification. In a single-dimensional game a weight is assigned to every transition and the objective of the protagonist is to assure a non-negative limit-average weight. In the multidimensional setting, a weight vector is assigned to every transition and the objective of the protagonist is to satisfy a boolean condition over the limit-average weight of each dimension, e.g., $\text{LimAvg}(x_1) \leq 0 \vee \text{LimAvg}(x_2) \geq 0 \wedge \text{LimAvg}(x_3) \geq 0$. We recently proved that when one of the players is restricted to finite-memory strategies then the decidability of determining the winner is inter-reducible with Hilbert’s Tenth problem over rationals (a fundamental long-standing open problem). In this work we consider arbitrary (infinite-memory) strategies for both players and show that the problem is undecidable.

1 Introduction

Two-player games on graphs provide the mathematical foundation for the study of reactive systems. In these games, the set of vertices is partitioned into player-1 and player-2 vertices; initially, a pebble is placed on an initial vertex, and in every round, the player who owns the vertex that the pebble resides in, advances the pebble to an adjacent vertex. This process is repeated forever and give rise to a *play* that induces an infinite sequence of edges. In the quantitative framework, an objective assigns a value to every play, and the goal of player 1 is to assure a value of at least ν to the objective. In order to have robust quantitative specifications, it is necessary to investigate games on graphs with multiple (and possibly conflicting) objectives. Typically, multiple objectives are modeled by multidimensional weight functions (e.g., $[4,5,7,1]$), and the outcome of a *play* is a vector of values (r_1, r_2, \dots, r_k) . A *robust specification* is a boolean formula over the atoms $r_i \sim \nu_i$, for $\sim \in \{\leq, <, \geq, >\}$, $i \in \{1, \dots, k\}$ and $\nu_i \in \mathbb{Q}$. For example, $\varphi = ((r_1 \geq 9 \vee r_2 \leq 9) \wedge r_3 < 0 \wedge r_4 > 9)$. The most well studied quantitative metric is the mean-payoff objective, which assigns the limit-average (long-run average) weight to an infinite sequence of weights (and if the limit does not exist, then we consider the limit infimum of the sequence). In this setting, r_i is the limit-average of dimension i of the weight function, and the goal of player 1 is to satisfy the boolean condition. In this work we prove that determining whether player 1 can satisfy such a condition is undecidable.

Related work. The model checking problem (one-player game) for such objectives (with some extensions) was considered in [1,6,3,11,12] and decidability was established. Two-player games for restricted subclasses that contain only conjunction of atoms were studied in [14,7,2,9] and tight complexity bounds were obtained (and in particular, the problem was proved to be decidable). In [15] a subclass that contains disjunction and conjunction of atoms of the form $r_i \sim \nu_i$ for $\sim \in \{\geq, >\}$ was studied and decidability was shown. In [13] we considered a similar objective but restricted player-1 to play only with finite-memory strategies. We showed that the problem is provably hard to solve and its decidability is inter-reducible with Hilbert’s tenth problem over rationals — a fundamental long standing open problem. In this work we consider for the first time games with robust quantitative class of specifications that is closed under boolean union, intersection and complement with arbitrary (infinite-memory) strategies.

Undecidability for (single-dimensional) mean-payoff games was proved for partial information mean-payoff games [10] and for mean-payoff games that are played over infinite-state pushdown automata [8]. These works did not exploit the different properties of the \geq and \leq operators (which correspond to the different properties of limit-infimum-average and limit-supremum-average). To the best of our knowledge, the undecidability proof in the paper is the first to exploit these properties. (As we mentioned before, when we consider only the \geq and $>$ operators, the problem is decidable.)

Structure of this paper. In the next section we give the formal definitions for robust multidimensional mean-payoff games. We prove undecidability by a reduction from the halting problem of a two-counter machine. For this purpose we first present a reduction from the halting problem of a one-counter machine and then we extend it to two-counter machine. In Section 3 we present the reduction and give an intuition about its correctness. In Section 4 we give a formal proof for the correctness of the reduction and extend the reduction to two-counter machine. In Section 5 we discuss the key elements of our proof and apply the undecidability result for the similar model of mean-payoff expressions.

2 Robust Multidimensional Mean-Payoff Games

Game graphs. A *game graph* $G = ((V, E), (V_1, V_2))$ consists of a *finite* directed graph (V, E) with a set of vertices V a set of edges E , and a partition (V_1, V_2) of V into two sets. The vertices in V_1 are *player-1 vertices*, where player 1 chooses the outgoing edges, and the vertices in V_2 are *player 2 vertices*, where player 2 (the adversary to player 1) chooses the outgoing edges. We assume that every vertex has at least one out-going edge.

Plays. A game is played by two players: player 1 and player 2, who form an infinite path in the game graph by moving a token along edges. They start by placing the token on an initial vertex, and then they take moves indefinitely in the following way. If the token is on a vertex in V_1 , then player 1 moves the token along one of the edges going out of the vertex. If the token is on a vertex

in V_2 , then player 2 does likewise. The result is an infinite path in the game graph, called *plays*. Formally, a *play* is an infinite sequence of vertices such that $(v_k, v_{k+1}) \in E$ for all $k \geq 0$.

Strategies. A strategy for a player is a rule that specifies how to extend plays. Formally, a *strategy* τ for player 1 is a function $\tau: V^* \cdot V_1 \rightarrow V$ that, given a finite sequence of vertices (representing the history of the play so far) which ends in a player 1 vertex, chooses the next vertex. The strategy must choose only available successors. The strategies for player 2 are defined analogously.

Multidimensional mean-payoff objectives. For multidimensional mean-payoff objectives we will consider game graphs along with a weight function $w: E \rightarrow \mathbb{Q}^k$ that maps each edge to a vector of rational weights. For a finite path π , we denote by $w(\pi)$ the sum of the weight vectors of the edges in π and $avg(\pi) = \frac{w(\pi)}{|\pi|}$, where $|\pi|$ is the length of π , denote the average vector of the weights. We denote by $avg_i(\pi)$ the projection of $avg(\pi)$ to the i -th dimension. For an infinite path π , let π_i denote the finite prefix of length i of π ; and we define $LimInfAvg_i(\pi) = \liminf_{i \rightarrow \infty} avg_i(\pi_i)$ and analogously $LimSupAvg_i(\pi)$ with \liminf replaced by \limsup . For an infinite path π , we denote by $LimInfAvg(\pi) = (LimInfAvg_1(\pi), \dots, LimInfAvg_k(\pi))$ (resp. $LimSupAvg(\pi) = (LimSupAvg_1(\pi), \dots, LimSupAvg_k(\pi))$) the limit-inf (resp. limit-sup) vector of the averages (long-run average or mean-payoff objectives). A multidimensional mean-payoff condition is a boolean formula over the atoms $LimInfAvg_i \sim \nu_i$ for $\sim \in \{\geq, >, \leq, <\}$. For example, the formula $LimInfAvg_1 > 8 \vee LimInfAvg_2 \leq -10 \wedge LimInfAvg < 9$ is a possible condition and a path π satisfies the formula if $LimInfAvg_1(\pi) > 8 \vee LimInfAvg_2(\pi) \leq -10 \wedge LimInfAvg(\pi) < 9$. We note that we may always assume that the boolean formula is positive (i.e., without negation), as, for example, we can always replace $\neg(r \geq \nu)$ with $r < \nu$.

For a given multidimensional weighted graph and a multidimensional mean-payoff condition, we say that player 1 is the winner of the game if he has a winning strategy that satisfy the condition against any player-2 strategy.

For an infinite sequence of reals x_1, x_2, x_3, \dots we have $LimInfAvg(x_1, x_2, \dots) = -LimSupAvg(-x_1, -x_2, \dots)$. Hence, an equivalent formulation for multidimensional mean-payoff condition is a positive boolean formula over the atoms $LimInfAvg_i \sim \nu_i$ and $LimSupAvg_i \sim \nu_i$ for $\sim \in \{\geq, >\}$. For positive formulas in which only the $LimInfAvg_i \sim \nu_i$ occur, determining the winner is decidable by [15]. In the sequel we abbreviate $LimInfAvg_i$ with \underline{i} and $LimSupAvg_i$ with \bar{i} . In this work we prove undecidability for the general case and for this purpose it is enough to consider only the \geq operator and thresholds 0. Hence, in the sequel, whenever it is clear that the threshold is 0, we abbreviate the condition $\underline{i} \geq 0$ with \underline{i} and $\bar{i} \geq 0$ with \bar{i} . For example, $\underline{i} \vee \bar{j} \wedge \underline{\ell}$ stands for $LimInfAvg_i \geq 0 \vee LimSupAvg_j \geq 0 \wedge LimInfAvg_\ell \geq 0$. By further abuse of notation we abbreviate the current total weight in dimension i by i (and make sure that the meaning of i is always clear from the context) and the absolute value of the total weight by $|i|$.

3 Reduction from the Halting Problem and Informal Proof of Correctness

In this work we prove the undecidability of determining the winner in games over robust mean-payoff condition by a reduction from the halting problem of two-counter machine. For this purpose we will first show a reduction from the halting problem of a one-counter machine to robust mean-payoff games, and the reduction from two-counter machines relays on similar techniques. We first give a formal definition for a one-counter machine, and in order to simplify the proofs we give a non-standard definition that is tailored for our needs. A *two-sided one-counter machine* M consists of two finite set of control states, namely Q (*left states*) and P (*right states*), an initial state $q_0 \in Q$, a final state $q_f \in Q$, a finite set of *left to right* instructions $\delta_{\ell \rightarrow r}$ and a finite set of *right to left* instructions $\delta_{r \rightarrow \ell}$. An instruction determines the next state and manipulates the value of the counter c (and initially the value of c is 0). A left to right instruction is of the form of either:

- $q : \text{if } c = 0 \text{ goto } p \text{ else } c := c - 1 \text{ goto } p'$, for $q \in Q$ and $p, p' \in P$; or
- $q : \text{goto } p$, for $q \in Q$ and $p \in P$ (the value of c does not change).

A right to left instruction is of the form of either

- $p : c := c + 1 \text{ goto } q$, for $p \in P$ and $q \in Q$; or
- $p : \text{goto } q$, for a state $p \in P$ and a state $q \in Q$ (the value of c does not change).

We observe that in our model, decrement operations are allowed only in left to right instructions and increment operations are allowed only in right to left instructions. However, since the model allows state transitions that do not change the value of the counter (*nop transitions*), it is trivial to simulate a standard one-counter machine by a two-sided counter machine.

For the reduction we use the states of the game graph to simulate the states of the counter machine and we use two dimensions to simulate the value of the one counter. In the most high level view our reduction consists of three main gadgets, namely, reset, sim and blame (see Figure 1), and a state q_f that represents the final state of the counter machine. Intuitively, in the sim gadget player 1 simulates the counter machine, and if the final state q_f is reached then player 1 loses. If player 2 detects that player 1 does not simulate the machine correctly, then the play goes to the blame gadget. From the blame gadget the play will eventually arrive to the reset gadget. This gadget assign proper values for all the dimensions of the game that are suited for an honest simulation in the sim gadget.

We now describe the construction with more details. We first present the winning objective and then we describe each of the three gadgets. For a two-sided counter machine M we construct a game graph with 8 dimensions denoted by $\ell, r, g_s, c_+, c_-, g_c, x$ and y and the objective

$$(\underline{\ell} \wedge \underline{r} \vee \overline{g_s}) \wedge (\underline{c_+} \wedge \underline{c_-} \vee \overline{g_c}) \wedge \overline{x} \wedge \overline{y}$$

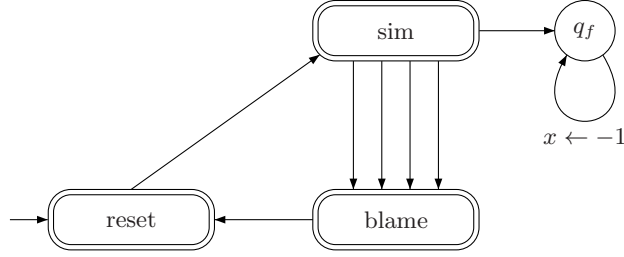


Fig. 1. Overview.

The sim gadget In the sim gadget player 1 suppose to simulate the run of M , and if the simulation is not honest, then player 2 activates the blame gadget. The simulation of the states is straight forward (since the game graph has states), and the difficulty is to simulate the counter value, more specifically, to simulate the zero testing of the counter. For this purpose we use the dimensions r, ℓ, g_s and c_+, c_-, g_c .

We first describe the role of r, ℓ and g_s . The reset gadget makes sure that in every invocation of the sim gadget, we have

$$\text{avg}(g_s) \approx -1, \text{avg}(r) \approx 1 \text{ and } \text{avg}(\ell) \approx 0$$

(The reader should read $a \approx b$ as "the value of a is *very close* to the value of b ". Precise definitions are given in Section 4) Then, during the simulation, the blame gadget makes sure that player 1 must play in such way that whenever the machine M is in a right state, then

$$r \approx |g_s| \text{ and } \ell \approx 0$$

and whenever the machine is in a left state, then

$$r \approx 0 \text{ and } \ell \approx |g_s|$$

Intuitively, the role of ℓ and r is to make sure that every left to right or right to left transition is simulated by a *significant* number of rounds in the sim gadget, and g_s is a *guard* dimension that make sure that the above assumptions on r and ℓ are satisfied.

We now describe the role of c_+, c_- and g_c . In the beginning of each simulation (i.e., every time that the sim gadget is invoked), we have

$$\text{avg}(c_+) \approx \text{avg}(c_-) \approx 1 \text{ and } \text{avg}(g_c) \approx -1$$

During the entire simulation we have $\text{avg}(g_c) \approx -1$ and if c is the value of the counter in the current simulation (i.e., since the sim gadget was invoked), then

$$c_+ \approx |g_c| + |g_s|c \text{ and } c_- \approx |g_c| - |g_s|c$$

Intuitively, whenever $c > 0$, then $c_- \lesssim |g_c|$, and if $c < 0$ (this can happen only if player 1 is dishonest), then $c_+ \lesssim |g_c|$.

We now describe the gadgets that simulate the operations of *inc*, *dec* and *nop*. The gadgets are illustrated in Figures 2-5 and the following conventions are used: (i) Player 1 owns the \circ vertices, player 2 owns the \square vertices, and the \square vertex stands for a gadget; (ii) A transition is labeled either with $a \leftarrow b$ symbol or with a text (e.g., "blame"). For a transition e the label $a \leftarrow b$ stands for $w_a(e) = b$. Whenever the weight of a dimension is not explicitly presented, then the weight is 0. We use text labels only to give intuition on the role of the transition. In such transitions the weights of all dimensions are 0.

In the first state of every *inc*, *dec* or *nop* gadget, in a left to right transition, player 1 loops until $\ell \approx 0$ and $r \approx g_s$, and in a right to left transition he loops until $r \approx 0$ and $\ell \approx g_s$. Then, player 2 decides whether he wants to blame player 1 for violating the assumptions about the values of ℓ , r and g_s .

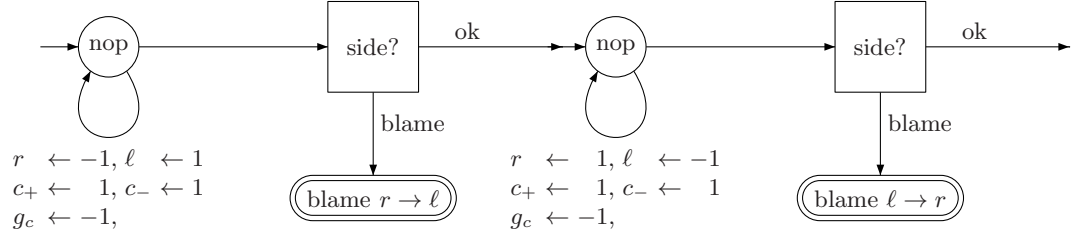


Fig. 2. *nop* $r \rightarrow \ell$ gadget.

Fig. 3. *nop* $\ell \rightarrow r$ gadget.

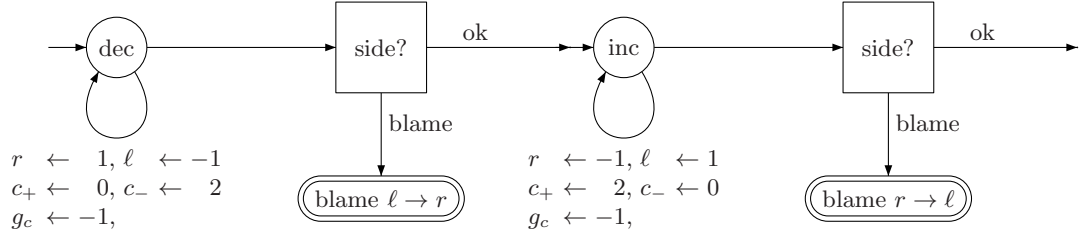


Fig. 4. *dec* $\ell \rightarrow r$ gadget.

Fig. 5. *inc* $r \rightarrow \ell$ gadget.

A transition $q : \text{if } c = 0 \text{ goto } p \text{ else } c := c - 1 \text{ goto } p'$, for $q \in Q$ and $p, p' \in P$ is described in Figure 6.

The blame gadgets The role of the blame gadgets is to make sure that the assumptions on ℓ , r and g_s are kept in the simulation and to make sure that the zero testing is honestly simulated. There are four blame gadgets. Two for the honest simulation of r , ℓ and g_s , and two for the zero testing (one for

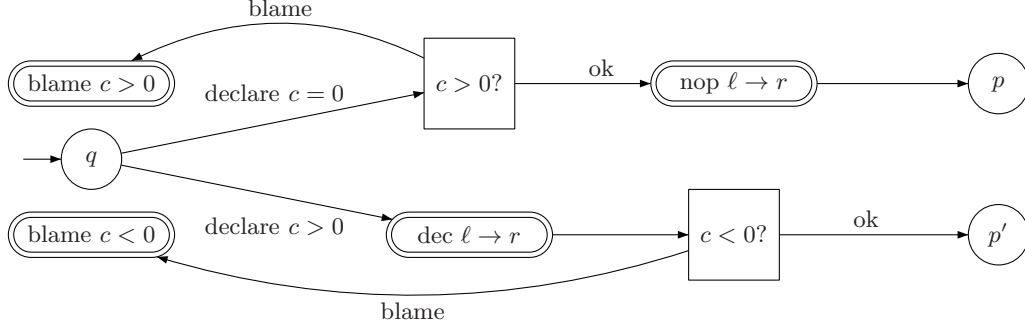


Fig. 6. q : if $c = 0$ then goto p else $c := c - 1$ goto p'

$c > 0$ and one for $c < 0$). The gadgets are described in Figures 8-10. The concept of all four gadgets is similar and hence we describe only the blame gadget for the left to right transition. We note that in an honest simulation we have $\text{avg}(r), \text{avg}(\ell), \text{avg}(c_+), \text{avg}(c_-), \text{avg}(x), \text{avg}(y) \gtrapprox 0$ in every round. Hence, if player 1 honestly simulates M and M does not halt, then the winning condition is satisfied. The blame gadget for the left to right transition is described in Figure 10. If the gadget is invoked and $r \not\gtrapprox g_s$, then player 2 can loop on the first state until $\text{avg}(r) \lesssim 0$ and still have $\text{avg}(g_s) \lesssim 0$. If $r \approx g_s$, then whenever we have $\text{avg}(r) \lesssim 0$ we will also have $\text{avg}(g_s) \gtrapprox 0$, and thus the winning objective is still satisfied. If $r \gtrapprox g_s$, then we must have $\text{avg}(\ell) \lesssim 0$. In this case, player 2 will immediately go to the reset gadget. We note that player 2 should eventually exit the blame gadget, since otherwise he will lose the game.

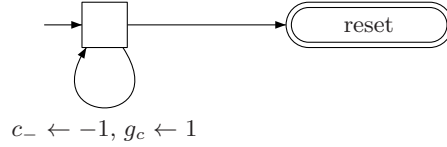


Fig. 7. blame $c > 0$ gadget.

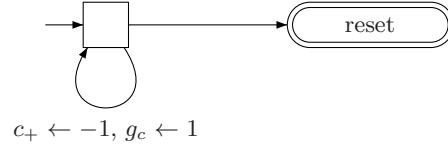


Fig. 8. blame $c < 0$ gadget.

The reset gadget The role of the reset gadget is to assign the following values for the dimensions:

$$\text{avg}(\ell) \approx 0, \text{avg}(r) \approx 1, \text{avg}(g_s) \approx -1, \text{avg}(c_-) \approx \text{avg}(c_+) \approx 1, \text{avg}(g_c) \approx -1$$

The gadget is described in Figure 11. We construct the gadget is such way that each of the players can enforce the above values (player 2 by looping enough

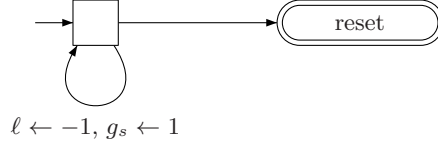


Fig. 9. blame $r \rightarrow \ell$ gadget.

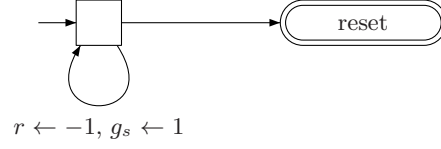


Fig. 10. blame $\ell \rightarrow r$ gadget.

time on the first state, player 1 by looping enough time on his two states). But the construction only gives this option to the players and it does not *punish* a player if he acts differently. However, the game graph is constructed in such way that if:

- M does not halt and in the reset gadget (at least one of the players) correctly reset the values, then player 1 wins.
- M halts and in the reset gadget (at least one of the players) correctly reset the values, then player 2 wins.

Hence, if M does halts, then player 2 winning strategy will make sure that the reset assigns correct values, and if M halts, then we can relay on player 1 to reset the values. We note that player 2 will not stay forever in his state (otherwise he will lose). In order to make sure that player 1 will not stay forever in one of his states we introduce two *liveness dimensions*, namely x and y . In the simulation and blame gadgets they get 0 values. But if player 1 remains forever in one of his two states in the reset gadget, then either x or y will have negative lim-sup value and player 1 will lose. Hence, in the reset gadget, player 1 should not only reset the values, but also assign a positive value for y and then a positive value for x .

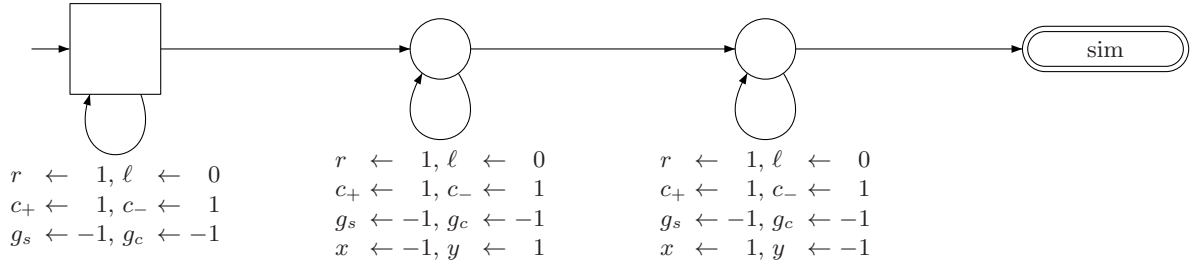


Fig. 11. Reset gadget.

Correctness of the reduction We claim that player 1 has a winning strategy if and only if the machine M does not halt. We first prove that if M halts, then player 2 has a winning strategy (the proof is informal, and a formal proof is given in Section 4). The winning strategy for player 2 is as follows: In the reset gadget

make sure that the *reset invariants* are satisfied. This is done by looping the first state of the reset gadget for enough rounds. In the sim gadget, whenever the *sim invariants* are not fulfilled or whenever player 1 cheats a zero-test, then player 2 invokes the blame gadget. If the sim invariants are fulfilled, then it must be the case that the game reaches state q_f , and in that case player 2 wins. Otherwise, in each simulation, the guard dimensions have negative average weights, while at least one of the dimensions ℓ, r, c_+ or c_- has a negative average weight in the blame gadget. Hence, we get that $\overline{g_s}, \overline{g_c} < 0$ and $\underline{\ell} < 0$ or $\underline{r} < 0$ or $\underline{c_-} < 0$ or $\underline{c_+} < 0$. Hence, the winning condition is not satisfied and player 2 is the winner.

To prove the converse direction we assume that M never halts and describe a winning strategy for player 1. The winning strategy is to honestly simulate M while keeping the *sim invariants* and the *reset invariants*. If player 2 never invokes the blame gadget, then the winning condition is trivially satisfied (since the game stays forever in the sim gadget). Otherwise, after every invocation of the blame gadget, if a *side blame gadget* was invoked, then either the average value of r and ℓ is non-negative or the value of the guard dimension g_s is non-negative. Hence, eventually, we get that $\underline{r}, \underline{\ell} \geq 0$ or $\overline{g_s} \geq 0$. And similarly, we get that $\underline{c_+}, \underline{c_-} \geq 0$ or $\overline{g_c} \geq 0$. Thus, the winning condition is satisfied, and player 1 is the winner.

4 Detailed Proof

In the previous section we accurately described the reduction, and only the proof of the correctness was informal. In this section we give a precise proof for the correctness of the reduction, namely, we formally describe player-2 winning strategy in the case that M halts (Subsection 4.1), and player-1 winning strategy in the case that M does not halt (Subsection 4.2). In Subsection 4.3 we extend the reduction to two-counter machine.

In the next subsections we use the next notations and definitions: A *round* is a round in the game graph (i.e., either player-1 or player-2 move). A *simulation step* denote all the rounds that are played in a transition gadget (i.e., in a *nop, inc* or *dec* gadget). The *total number of rounds* is the total number of rounds from the beginning of the play.

4.1 If M halts, then player 2 is the winner

In this subsection we assume that M halts. We denote by N the number of steps after which M halts (for initial counter value 0) and we denote $\epsilon = \frac{1}{(N+1)^2}$. WLOG we assume that $N > 10$. The strategy of player 2 in the reset gadget is to achieve the following *reset invariants* (after the play leaves the gadget):

- $avg(g_s), avg(g_c) \leq -\frac{1}{2}$
- $(1 - \frac{\epsilon}{4})|g_s| \leq r \leq (1 + \frac{\epsilon}{4})|g_s|$
- $-\frac{\epsilon}{4}|g_s| \leq \ell \leq \frac{\epsilon}{4}|g_s|$
- $(1 - \frac{\epsilon}{4})|g_c| \leq c_+, c_- \leq (1 + \frac{\epsilon}{4})|g_c|$

We note that player 2 can maintain the above by looping sufficiently long time in the first state, and once the invariants are reached, player 1 cannot violate them in his states in the reset gadget (since the average value of g_s and g_c can only get closer to -1 , the average value of ℓ only gets closer to 0 and the average value of r, c_- and c_+ only gets closer to 1).

The strategy of player 2 in the sim gadget is to maintain, in every step of the simulation, the next two invariants, which we denote by the *left right invariants*:

- (Left state invariant) If the machine is in a left state, then $(1 - \epsilon)|g_s| \leq \ell \leq (1 + \epsilon)|g_s|$ and $-\epsilon|g_s| \leq r \leq \epsilon|g_s|$.
- (Right state invariant) If the machine is in a right state, then $(1 - \epsilon)|g_s| \leq r \leq (1 + \epsilon)|g_s|$ and $-\epsilon|g_s| \leq \ell \leq +\epsilon|g_s|$

We denote $\delta = \frac{1}{\frac{1}{2} + N(1+2\epsilon)}$. We first prove that under these invariants $avg(g_s) \leq -\delta$. Then we use this fact to show that if player 1 violates these invariants, then player 2 can violate $(\underline{\ell} \wedge \underline{r} \vee \overline{g_s})$, and therefore he wins.

Lemma 1 *Suppose that after the reset gadget $avg(g_s) \leq -\frac{1}{2}$, then as long as the left right invariants are maintained, it always hold that $avg(g_s) \leq -\delta$ (provided that the sim gadget does not simulate more than N steps).*

Proof. We denote by R the number of rounds that were played before the current invocation of the simulation gadget. We claim that after simulating i steps of the machine (in the current invocation of the sim gadget), the **total** number of rounds in the play (i.e., number of rounds from the beginning of the play, not from the beginning of the current invocation) is at most $R + i \cdot |g_s|(1 + 2\epsilon)$. The proof is by a simple induction, and for the base case $i = 0$ the proof is trivial. For $i > 0$, we assume WLOG that the i -th transition is a left-to-right transition. Hence, before the last simulation step we had $r \geq -\epsilon|g_s|$ and after the i -th step was completed we have $r \leq (1 + \epsilon)|g_s|$. Since in every round the value of r is incremented by 1, we get that at most $(1 + 2\epsilon)|g_s|$ rounds were played and the proof of the claim follows (and the proof for a right-to-left transition is symmetric).

Hence, after N simulation steps we have

$$avg(g_s) \leq \frac{g_s}{R + N|g_s|(1 + 2\epsilon)}$$

Since in the beginning of the sim gadget we had $avg(g_s) \leq -\frac{1}{2}$, then $R \leq \frac{|g_s|}{2}$. Hence, and since $g_s < 0$ we get

$$avg(g_s) \leq \frac{g_s}{\frac{|g_s|}{2} + N|g_s|(1 + 2\epsilon)} = -\frac{1}{\frac{1}{2} + N(1 + 2\epsilon)} = -\delta$$

We note that unless the blame gadget is invoked, the value of g_s is not changed in the simulation gadget. Hence, $avg(g_s)$ gets the maximal value after the N -th step and the proof is complete. \square

Lemma 2 *Let $\gamma = \min(\frac{\epsilon\delta}{2}, \frac{\frac{\epsilon}{4}}{1+\frac{1}{\delta}-\frac{\epsilon}{4}})$. If player 1 violates the left-right invariants in the first N steps of a simulation, then player 2 can achieve in the blame gadget either $\text{avg}(r) \leq -\gamma$ or $\text{avg}(\ell) \leq -\gamma$ (or both) while maintaining $\text{avg}(g_s), \text{avg}(g_c) \leq -\gamma$.*

Proof. We prove the assertion for the left-state invariant and the proof for the right-state is symmetric. Recall that the invariant consistences of four assumptions, namely, (i) $(1 - \epsilon)|g_s| \leq \ell$; (ii) $\ell \leq (1 + \epsilon)|g_s|$; (iii) $-\epsilon|g_s| \leq r$; and (iv) $r \leq \epsilon|g_s|$. We first prove the assertion when the first condition is violated, i.e., we assume that $\ell < (1 - \epsilon)|g_s|$. If this is the case after a right-to-left transition, then player 2 will invoke the blame $r \rightarrow \ell$ gadget after the transition ends. In the blame gadget he will traverse the self-loop for $X \cdot (1 - \frac{\epsilon}{2})$ times, where X is the value of $|g_s|$ before the invocation of the blame gadget, and then he will go to the reset gadget. As a result (since in every loop ℓ is decremented by 1 and g_s is incremented by 1) we get that the value of ℓ and g_s is at most $-X \cdot \frac{\epsilon}{2}$. By Lemma 1 we know that before the invocation of the blame gadget we had $\text{avg}(g_s) \leq -\delta$. Hence, if R is the number of rounds before the invocation of the blame gadget, then $R \leq \frac{X}{\delta}$. Hence, after the blame gadget ends, we have

$$\text{avg}(\ell), \text{avg}(g_s) \leq -\frac{X \cdot \frac{\epsilon}{2}}{R + X \cdot (1 - \frac{\epsilon}{2})} \leq -\frac{X \cdot \frac{\epsilon}{2}}{\frac{X}{\delta} + X \cdot (1 - \frac{\epsilon}{2})} = -\frac{\frac{\epsilon}{2}}{1 + \frac{1}{\delta} - \frac{\epsilon}{2}}$$

If the second condition is violated, namely, if $\ell > (1 + \epsilon)|g_s|$, then we claim that it must be the case that $r < -\frac{\epsilon|g_s|}{2}$. Indeed, when the sim gadget is invoked we have $r \leq |g_s|(1 + \frac{\epsilon}{4})$ and $\ell \leq |g_s|\frac{\epsilon}{4}$. In the sim gadget the value of the sum $r + \ell$ is not changed (since r is incremented only when ℓ is decremented and vice versa). Hence, the sum never exceeds $|g_s|(1 + \frac{\epsilon}{2})$. Thus, if $\ell > (1 + \epsilon)|g_s|$, then it must be the case that $r < -\frac{\epsilon|g_s|}{2}$ and we get that $\text{avg}(r) \leq -\frac{\epsilon}{2}\delta$. Hence, by leaving the blame gadget after two rounds we get $\text{avg}(r) \leq -\frac{\epsilon\delta}{2}$ while $\text{avg}(g_s) \leq -\delta$.

If the third condition is violated, namely, if $r < -\epsilon|g_s|$, then $\text{avg}(r) = -\epsilon|\text{avg}(g_s)| \leq -\epsilon\delta$. Hence, if player 2 can choose to exit the blame gadget after two rounds and we get $\text{avg}(r) \leq -\epsilon\delta$ while $\text{avg}(g_s) \leq -\delta$.

Finally, if the fourth condition is violated, namely, if $r > \epsilon|g_s|$, then by analyzing the sum $r + \ell$ we get that $\ell \leq (1 - \frac{\epsilon}{2})|g_s|$. We repeat the same analysis as in the case where the first invariant is violated (i.e., when $\ell \leq (1 - \epsilon)|g_s|$) and get that

$$\text{avg}(g_s), \text{avg}(r) \leq -\frac{\frac{\epsilon}{4}}{1 + \frac{1}{\delta} - \frac{\epsilon}{4}}$$

It is an easy observation that if at beginning of the last invocation of the sim gadget $\text{avg}(g_c) \leq -\frac{1}{2}$, then it remains at most $-\frac{1}{2}$ as it gets a value of -1 in every round. The proof is complete. \square

By Lemma 2, if player 2 maintains the reset invariant in the reset gadget, then in every simulation player 1 must satisfy the left-right invariants. Otherwise, we get that infinitely often the average value of either r or ℓ is $-\gamma$ while the average

value of g_s is always at most $-\gamma$. Hence $\overline{g_s} < 0$ and either $\underline{r} < 0$ or $\underline{\ell} < 0$ and thus the condition $(\underline{\ell} \wedge \underline{r} \vee \overline{g_s})$ is violated and therefore player 1 is losing.

In the next three lemmas we prove that player 1 must honestly simulates the zero-testing. The first lemma is a simple corollary from the left-right invariants.

Lemma 3 *Under the left-right invariants, in the dec, inc and nop gadgets, player 1 follows the self-loop of the first state at most $|g_s|(1 + 2\epsilon)$ rounds and at least $|g_s|(1 - 2\epsilon)$ rounds.*

Proof. Upper bound: in a left to right transition, we have $\ell \leq |g_s|(1 + \epsilon)$ before the transition and $\ell \geq -|g_s|\epsilon$ after the transition, and in every round ℓ is decremented by 1. Lower bound: in a left to right transition, we have $\ell \geq |g_s|(1 - \epsilon)$ before the transition and $\ell \leq |g_s|\epsilon$ after the transition, and in every round ℓ is decremented by 1.

The proof for a right to left transition is symmetric.

The next lemma shows the correlation between g_c and c_+ and c_- .

Lemma 4 *Let $\#inc$ (resp., $\#dec$) be the number of times that the inc (dec) gadget was visited, and we denote $c = \#inc - \#dec$ (namely, c is the actual value of the counter in the counter machine M). Then under the left-right invariants, in the first N steps of the simulation we always have $c_+ \leq |g_c|(1 + \epsilon) + c|g_s| + \frac{|g_s|}{2}$ and $c_- \leq |g_c|(1 + \epsilon) - c|g_s| + \frac{|g_s|}{2}$.*

Proof. We prove the claim of the lemma for c_+ and the proof for c_- is symmetric. Let X be the value of $|g_c|$ when the sim gadget is invoked. By the reset invariants we get that $c_+ \leq X(1 + \frac{\epsilon}{4})$. By Lemma 3 we get that every visit in the inc gadget contributes at most $|g_s|(1 + 2\epsilon)$ more to c_+ than its contribution to g_c and every visit in the dec contributes at least $|g_s|(1 - 2\epsilon)$ more to g_c than its contribution to c_+ . Hence,

$$c_+ \leq X(1 + \frac{\epsilon}{4}) + (|g_c| - X) + \#inc \cdot |g_s|(1 + 2\epsilon) - \#dec \cdot |g_s|(1 - 2\epsilon) = |g_c| + \epsilon X + (\#inc - \#dec)|g_s|(1 + 2\epsilon) + 4\epsilon|g_s| \cdot \#dec$$

We recall that $c = (\#inc - \#dec)$, and observe that $X \leq |g_c|$, and that $\#dec \leq N$ and thus $\epsilon \cdot \#dec < \frac{1}{10}$. Hence, we get that $c_+ \leq |g_c|(1 + \epsilon) + c|g_s| + \frac{|g_s|}{2}$. \square

The next lemma suggest that player 1 must honestly simulates the zero-tests.

Lemma 5 *If the reset and left-right invariants hold, then for $\gamma = \min(\frac{1}{20N}, \frac{\delta}{8})$ the following hold: (i) if the blame $c < 0$ is invoked and $c < 0$ then player 2 can achieve $c_+ \leq -\gamma$ while maintaining $\text{avg}(g_s), \text{avg}(g_c) \leq -\gamma$; and (ii) if the blame $c > 0$ is invoked and $c > 0$ then player 2 can achieve $c_- \leq -\gamma$ while maintaining $\text{avg}(g_s), \text{avg}(g_c) \leq -\gamma$.*

Proof. We prove the first item of the lemma and the proof for the second item is symmetric. Suppose that $c < 0$ (i.e., $c \leq -1$) when blame $c < 0$ gadget is invoked. Let X and Y be the values of $|g_c|$ and $|g_s|$ before the invocation of the blame gadget. Then by Lemma 4, before the invocation we have $c_+ \leq X(1 + \epsilon) - \frac{Y}{2}$. Hence, by traversing the self-loop in the first state of the blame $c < 0$ gadget for $X(1 + \epsilon) - \frac{Y}{4}$ rounds we get $c_+ \leq -\frac{Y}{4}$ and $g_c \leq \epsilon X - \frac{Y}{4}$. Let R be the number of rounds that were played from the beginning of the play (and not just from the beginning of the current invocation of the sim gadget). Since g_c is decremented by at most 1 in every round we get that $X(1 + \epsilon) - \frac{Y}{4} \leq 2X \leq 2R$. By lemma 1 we have $\frac{Y}{R} \leq -\delta$. Hence, $\text{avg}(c_+) \leq \frac{c_+}{2R} \leq -\frac{Y}{8R} \leq -\frac{\delta}{8}$. Similarly, since $\frac{X}{R}$ is bounded by 1, we have $\text{avg}(g_c) \leq \frac{\epsilon X}{2R} - \frac{\delta}{8} \leq \frac{\epsilon}{2} - \frac{\delta}{8}$. Recall that $\delta = \frac{1}{\frac{1}{2} + N(1 + 2\epsilon)}$. Hence, $\text{avg}(g_c) \leq \frac{2\epsilon + 4N\epsilon + 8\epsilon^2 - 1}{8(\frac{1}{2} + N(1 + 2\epsilon))}$ and since $\epsilon = \frac{1}{(N+1)^2}$ and $N > 10$ we get that $\text{avg}(g_c) \leq -\frac{1}{20N}$. The value of g_s was at most $-\delta R$ before the blame gadget, and in the blame gadget g_s is decreased by 1 in every round. Hence $\text{avg}(g_s) \leq -\delta$ and the proof follows by taking $\gamma = \min(\frac{1}{20N}, \frac{\delta}{8})$. \square

We are now ready to prove one side of the reduction.

Proposition 1 *If the counter machine M halts, then player 2 has a winning strategy for violating $(\underline{\ell} \wedge \underline{r} \vee \overline{g_s}) \wedge (\underline{c_+} \wedge \underline{c_-} \vee \overline{g_c}) \wedge \overline{x} \wedge \overline{y}$. Moreover, if M halts then there exists a constant $\zeta > 0$ that depends only on M such that player 2 has a winning strategy for violating $(\underline{\ell} \geq -\zeta \wedge \underline{r} \geq -\zeta \vee \overline{g_s} \geq -\zeta) \wedge (\underline{c_+} \geq -\zeta \wedge \underline{c_-} \geq -\zeta \vee \overline{g_c} \geq -\zeta) \wedge \overline{x} \geq -\zeta \wedge \overline{y} \geq -\zeta$.*

Proof. Suppose that M halts and let N be the number of steps that M runs before it halts (for an initial counter value 0). Player-2 strategy is to

- Maintain the reset-invariants.
- Whenever the left-right invariants are violated, he invokes the blame $r \rightarrow \ell$ or $\ell \rightarrow r$ gadget.
- Whenever the zero-testing is dishonest, he activates the corresponding blame gadget (either $c > 0$ or $c < 0$).
- If q_f is reached, he stays there forever.

The correctness of the construction is immediate by the lemmas above. We first observe that it is possible for player 2 to satisfy the reset-invariants and that if player 1 stays in the reset gadget forever, then he loses.

Whenever the left-right invariant is violated, then the average weight of r and/or ℓ is negative, while the average weight of g_s and g_c remains negative. Hence, if in every simulation player 1 violates the left-right invariants in the first N steps we get that the condition is violated since $\overline{g_s} \leq -\gamma$ and either $\underline{r} \leq -\gamma$ or $\underline{\ell} \leq -\gamma$. Hence, we may assume that these invariants are kept in every simulation.

Whenever the zero-testing is dishonest (while the left-right invariants are satisfied), then by Lemma 5, player 2 can invoke a counter blame gadget and achieve negative average for either c_+ or c_- while maintaining g_c and g_s negative.

If in every simulation player 1 is dishonest in zero-testing, then we get that either $\underline{c}_- \leq -\gamma$ or $\underline{c}_+ \leq -\gamma$ while $\overline{g}_c \leq -\gamma$ and the condition is violated. Hence, we may assume that player 1 honestly simulates the zero-tests. Finally, if the transitions of M are properly simulated, then it must be the case the state q_f is reached and when looping this state forever player 1 loses (since $\overline{x} \leq -1 < 0$). \square

4.2 If M does not halt, then player 1 is the winner

Suppose that M does not halt. The strategy of player 1 in the reset gadget is as following: Let i be the number of times that the reset gadget was visited, and we denote $\epsilon_i = \frac{1}{i+10}$. Similarly to player-2 strategy in Subsection 4.1, player-1 strategy in the reset gadget is to achieve the following invariants (after the play leaves the gadget):

$$\begin{aligned} & - \text{avg}(g_s), \text{avg}(g_c) \leq -\frac{1}{2} \\ & - (1 - \frac{\epsilon_i}{4})|g_s| \leq r \leq (1 + \frac{\epsilon_i}{4})|g_s| \\ & - -\frac{\epsilon_i|g_s|}{4} \leq \ell \leq \frac{\epsilon_i|g_s|}{4} \\ & - (1 - \frac{\epsilon_i}{4})|g_c| \leq c_+, c_- \leq (1 + \frac{\epsilon_i}{4})|g_c| \end{aligned}$$

To satisfy these invariants, he follows the self-loop of his first state until $\text{avg}(y) \geq 0$ and then follows the self-loop of the second state until the invariants are fulfilled and $\text{avg}(x) \geq 0$. In the sim gadget, player-1 strategy is to simulate every *nop*, *inc* and *dec* step by following the self-loop in the corresponding gadget for $|g_s|$ rounds. In addition, player 1 honestly simulates the zero-tests.

We denote the above player-1 strategy by τ and next lemma shows the basic properties of a play according to τ .

Lemma 6 *In any play according to τ , after the reset gadget was visited for i times, in the sim gadget we always have: (i) In a right state: $r \geq -\epsilon_i|g_s|$, $\ell \geq (1 - \epsilon_i)|g_s|$ and in a left state $\ell \geq -\epsilon_i|g_s|$, $r \geq (1 - \epsilon_i)|g_s|$; and (ii) $c_+ \geq (1 - \epsilon_i)|g_c| + c|g_s|$ and $c_- \geq (1 - \epsilon_i)|g_c| - c|g_s|$, where $c = \#inc - \#dec$ in the current invocation of the sim gadget.*

Proof. The proof of the first item is straight forward. Initially, when the sim gadget is invoked (and starts in a left state) we have $-\epsilon_i|g_s| \leq \ell \leq \epsilon_i|g_s|$ and $(1 - \epsilon_i)|g_s| \leq r \leq (1 + \epsilon_i)|g_s|$. In every left to right transition r is decremented by $|g_s|$ and ℓ is incremented by $|g_s|$. In every right to left transition ℓ is decremented by $|g_s|$ and r is incremented by $|g_s|$. Hence, the proof of the first item follows.

In order to prove the second item we denote by K the value of $|g_c|$ when the sim gadget is invoked. Initially we have $c_+ \geq (1 - \epsilon_i)K$. After every simulation step we have $c_+ \geq (1 - \epsilon_i)K + |g_s|(2\#inc + \#nop)$ and $|g_c| = K + |g_s|(\#inc + \#dec + \#nop)$. Hence, $c_+ \geq (1 - \epsilon_i)|g_c| + c|g_s|$. The proof for c_- is symmetric. \square

We will use the next lemma to prove that player 2 is losing when he invokes the blame gadgets.

Lemma 7 *In a play prefix consistent with τ :*

1. In the blame $r \rightarrow \ell$ gadget: whenever $\text{avg}(\ell) \leq -\epsilon_i$, then $\text{avg}(g_s) \geq -\epsilon_i$.
2. In the blame $\ell \rightarrow r$ gadget: whenever $\text{avg}(r) \leq -\epsilon_i$, then $\text{avg}(g_s) \geq -\epsilon_i$.
3. In the blame $c < 0$ gadget: whenever $\text{avg}(c_+) \leq -\epsilon_i$, then $\text{avg}(g_c) \geq -\epsilon_i$.
4. In the blame $c > 0$ gadget: whenever $\text{avg}(c_-) \leq -\epsilon_i$, then $\text{avg}(g_c) \geq -\epsilon_i$.

Where i is the number of times that the reset gadget was visited.

Proof. Proof of item 1: By Lemma 6 we know that when blame $r \rightarrow \ell$ gadget is invoked, the value of ℓ is at least $(1 - \epsilon_i)|g_s|$. Let R be the total number of rounds that were played before the blame gadget was invoked, and by Y the value of $|g_s|$ before the invocation of the blame gadget. In order to obtain $\text{avg}(\ell) \leq -\epsilon_i$, player 1 should follow the self-loop of the gadget at least $X = \frac{(1 - \epsilon_i)Y + R\epsilon_i}{1 - \epsilon_i}$ rounds. In every such round the value of g_s is incremented by 1. We note that $X \geq Y$. Hence, after X rounds we have $g_s = -Y + X \geq 0$ and in particular $\text{avg}(g_s) \geq -\epsilon_i$.

Proof of item 2: Symmetric to the proof of item 1.

Proof of item 3: By Lemma 7 we know that before the invocation of the blame gadget we have $c_+ \geq (1 - \epsilon_i)|g_c| + c|g_s|$. Since τ honestly simulates the zero-tests we get that $c \geq 0$. Hence, $c_+ \geq (1 - \epsilon_i)|g_c|$. Thus, by the same arguments as in the proof of item 1 we get that if $\text{avg}(c_+) \leq -\epsilon_i$, then $\text{avg}(g_c) \geq 0 \geq -\epsilon_i$.

Proof of item 4: By Lemma 7 we know that before the invocation of the blame gadget we have $c_- \geq (1 - \epsilon_i)|g_c| - c|g_s|$. Since τ honestly simulates the zero-tests we get that $c = 0$. The rest of the proof is symmetric to the proof of item 3. \square

We are now ready to prove the τ is a winning strategy.

Proposition 2 *If M does not halt, then τ is a winning strategy.*

Proof. In order to prove that τ satisfies the condition $(\underline{\ell} \wedge \underline{r} \vee \overline{g_s}) \wedge (c_+ \wedge c_- \vee \overline{g_c}) \wedge \overline{x} \wedge \overline{y}$ it is enough to prove that when playing according to τ , for any constant $\delta > 0$ the condition $(\underline{\ell} \geq -\delta \wedge \underline{r} \geq -\delta \vee \overline{g_s} \geq -\delta) \wedge (c_+ \geq -\delta \wedge c_- \geq -\delta \vee \overline{g_c} \geq -\delta) \wedge \overline{x} \wedge \overline{y}$ is satisfied.

Let $\delta > 0$ be an arbitrary constant and in order to prove the claim we consider two distinct cases.

In the first case, player 2 strategy will invoke the blame gadgets only finitely many times. Hence, either there is a suffix that is played only in a blame gadget or only in a reset gadget (and in such suffixes player 2 loses) or there is an infinite suffix that is played only in the sim gadget (as M does not halt). In the sim gadget the values of x, y and g_s are not changed. Hence the long-run average weight of these dimensions is 0. In addition, c_+ and c_- are never decremented in the sim gadget. Hence, their long-run average weight is also at least 0. Thus, the condition is satisfied.

In the second case we consider, player 2 always eventually invokes a blame gadget. Since a blame gadget is invoked infinitely many times we get that the reset gadget is invoked infinitely often, and thus $\overline{x}, \overline{y} \geq 0$. In addition, the sim gadget is invoked infinitely often. Let i be the minimal index for which $\epsilon_i \leq \delta$. We

claim that after the i -th invocation of the sim gadget, in every round (i) either $avg(\ell) \geq -\epsilon_i \wedge avg(r) \geq -\epsilon_i$ or $avg(g_s) \geq -\epsilon_i$; and (ii) either $avg(c_+) \geq -\epsilon_i \wedge avg(c_-) \geq -\epsilon_i$ or $avg(g_c) \geq -\epsilon_i$. The proof for the first item follows from the reset invariants, from Lemma 6 (as $r, \ell \geq -\epsilon_i |g_s|$ and $|avg(g_s)| \leq 1$) and from Lemma 7 (and from the fact that r and ℓ are never decremented in the reset gadget). The proof for the second item follows from the reset invariants, from the fact that c_+ and c_- are never decremented in the sim gadget and from Lemma 7 (and from the fact that c_+ and c_- are never decremented in the reset gadget). Thus, from certain round, either $avg(\ell)$ and $avg(r)$ are always at least $-\delta$, or infinitely often $avg(g_s) \geq -\delta$. Hence, $(\underline{\ell} \geq -\delta \wedge \underline{r} \geq -\delta \vee \overline{g_s} \geq -\delta)$ is satisfied and similarly $(\underline{c_+} \geq -\delta \wedge \underline{c_-} \geq -\delta \vee \overline{g_c} \geq -\delta)$ is satisfied. Therefore, since $\epsilon_i \leq \delta$, we get that τ satisfies $(\underline{\ell} \geq -\delta \wedge \underline{r} \geq -\delta \vee \overline{g_s} \geq -\delta) \wedge (\underline{c_+} \geq -\delta \wedge \underline{c_-} \geq -\delta \vee \overline{g_c} \geq -\delta) \wedge \bar{x} \wedge \bar{y}$. \square

4.3 Extending the reduction to two-counter machine

When M is a two-counter machine, we use 4 dimensions for the counters, namely $c_+^1, c_+^2, c_-^1, c_-^2$ and one guard dimension g_c . The winning condition is $(\underline{\ell} \wedge \underline{r} \vee \overline{g_s}) \wedge (\underline{c_+^1} \wedge \underline{c_+^2} \wedge \underline{c_-^1} \wedge \underline{c_-^2} \vee \overline{g_c}) \wedge \bar{x} \wedge \bar{y}$. In a *nop* gadget all four dimensions $c_+^1, c_+^2, c_-^1, c_-^2$ get a value of 1 in the self-loop. When a counter c_i (for $i = 1, 2$) is incremented (resp., decremented), then counter c_+^i and c_-^i are assigned with weights according to the weights of c_+ and c_- in the *inc* (*dec*) gadget that we described in the reduction for a one counter machine, and c_+^{3-i}, c_-^{3-i} are assigned with weights according to a *nop* gadget.

The proofs of Proposition 1 and Proposition 2 easily scale to a two-counter machine. Hence, the undecidability result is obtained.

Theorem 1 *The problem of deciding the winner in a robust multidimensional mean-payoff game with ten dimensions is undecidable.*

5 Discussion

Additional intuition. Previous undecidability results for mean-payoff games relied on a reduction from the universality problem of a non-deterministic weighted automaton over finite-words. In such setting, the input for the automaton is a run of a counter machine and the automaton verifies that the run is valid. The non-determinism allows the automaton to decide after the run ends whether (i) it simulates the first counter or the second counter; and (ii) whether a dishonest zero-test was done for a positive counter or for a zero counter.

In our reduction player 2 must identify dishonest simulation during the run and he does not know in advance whether a dishonest zero-test will be done for the first counter or for the second counter. Hence, if we would apply the standard technique, then one run that is dishonest for the first counter might *fix* the mean-payoff of a previous run that was dishonest for the second counter. For this purpose we introduce the reset gadget that in a way *clears* the affect of

previous simulations. However, with the reset gadget the affect of the rounds that are played in the simulation gadget becomes neglectable. For this purpose we introduce the notion of left-to-right and right-to-left transitions who make sure that every simulation step takes a non-neglectable fraction of the run. Both the reset gadget and the blame gadgets crucially relies on the combination of limit-supremum-average and limit-infimum-average objectives. Indeed, games over the condition $(\underline{\ell} \wedge \underline{r} \vee \underline{g_s}) \wedge (\underline{c_+^1} \wedge \underline{c_-^1} \wedge \underline{c_+^2} \wedge \underline{c_-^2} \vee \underline{g_c}) \wedge \underline{x} \wedge \underline{y}$ or the condition $(\overline{\ell} \wedge \overline{r} \vee \overline{g_s}) \wedge (\overline{c_+^1} \wedge \overline{c_-^1} \wedge \overline{c_+^2} \wedge \overline{c_-^2} \vee \overline{g_c}) \wedge \overline{x} \wedge \overline{y}$ are decidable [15].

Undecidability for Similar Objectives Alur et al [1] considered the same objectives as in this work. Boker et al. [3] and Tomita et al. [11] extended Alur et al objectives with boolean temporal logic. Hence, the undecidability result trivially holds for their model.

Chatterjee et al. [6] considered a quantitative objective that assigns a (one-dimensional) real value to the vectors $LimInfAvg(\pi)$ and $LimSupAvg(\pi)$. In their framework an atomic expression is $LimInfAvg_i$ and $LimSupAvg_i$ and a *mean-payoff expression* is the closure under the min, max and sum operators and under the numerical complement (multiplication by -1). For example, the value of the expression $E = LimInfAvg_1 + \max(LimInfAvg_2, LimSupAvg_3)$ for an infinite path π is $LimInfAvg_1(\pi) + \max(LimInfAvg_2(\pi), LimSupAvg_3(\pi))$. As the objective is quantitative and not boolean, there are three problem that are relevant to games for a given expression E : (i) Can player 1 assure $E \geq \nu$?; (ii) Can player 1 assure $E > \nu$?; and (iii) What is the maximal (supremum) value the player 1 can assure? In [13] we proved that when player 1 is restricted to finite-memory strategies, then the decidability of the first problem is inter-reducible with Hilbert's Tenth problem over rationals (a long standing open problem), but the second problem is decidable and the third problem is computable. For arbitrary (infinite-memory) strategies we can easily extend our undecidability proof from Section 3 and 4 and get the next theorem:

Theorem 2 *Two-player games over mean-payoff expression are undecidable.*

Proof. Consider the construction from Section 3 and the expressions

$$E_1 = \max(\min(LimInfAvg_{\ell}, LimInfAvg_r), LimSupAvg_{g_s}),$$

$$E_2 = \max(\min(LimInfAvg_{c_+^1}, LimInfAvg_{c_-^1}, LimInfAvg_{c_+^2}, LimInfAvg_{c_-^2}), LimSupAvg_{g_c}),$$

$$E_3 = \min(LimSupAvg_x, LimSupAvg_y), E = \min(E_1, E_2, E_3) \text{ and } F = -E, \text{ and}$$

the boolean formula $\varphi = (\underline{\ell} \wedge \underline{r} \vee \underline{g_s}) \wedge (\underline{c_+^1} \wedge \underline{c_-^1} \wedge \underline{c_+^2} \wedge \underline{c_-^2} \vee \underline{g_c}) \wedge \underline{x} \wedge \underline{y}$. It is a trivial observation that for any play π : (i) φ is satisfied iff $E(\pi) \geq 0$; and (ii) φ is not satisfied iff $F(\pi) > 0$. Hence, we have a reduction from determining whether player 1 wins for robust multidimensional mean-payoff games to mean-payoff expression games with weak inequality, and a reduction from determining whether player 2 wins for robust multidimensional mean-payoff games to mean-payoff expression games with strong inequality. Moreover, by Proposition 1 it follows that if player 1 cannot satisfy $E \geq 0$, then for some $\zeta > 0$ he cannot satisfy $E \geq -\zeta$. Hence if we could compute the maximal (supremum) value that

player 1 can assure we could determine whether he wins for $E \geq 0$ (if the value is negative, then he loses, and otherwise he wins). \square

References

1. R. Alur, A. Degorre, O. Maler, and G. Weiss. On omega-languages defined by mean-payoff conditions. *FOSSACS*, 2009.
2. A. Bohy, V. Bruyère, E. Filiot, and J.-F. Raskin. Synthesis from ltl specifications with mean-payoff objectives. In *TACAS*, 2013.
3. U. Boker, K. Chatterjee, T. A. Henzinger, and O. Kupferman. Temporal specifications with accumulative values. In *LICS*, 2011.
4. T. Brázdil, V. Brozek, K. Chatterjee, V. Forejt, and A. Kucera. Two views on multiple mean-payoff objectives in markov decision processes. In *LICS*, 2011.
5. T. Brázdil, K. Chatterjee, A. Kucera, and P. Novotný. Efficient controller synthesis for consumption games with multiple resource types. In *CAV*, 2012.
6. K. Chatterjee, L. Doyen, H. Edelsbrunner, T. A. Henzinger, and P. Rannou. Mean-payoff automaton expressions. In *CONCUR*, 2010.
7. K. Chatterjee, M. Randour, and J.-F. Raskin. Strategy synthesis for multidimensional quantitative objectives. In *CONCUR*, 2012.
8. K. Chatterjee and Y. Velner. Mean-payoff pushdown games. In *LICS*, 2012.
9. K. Chatterjee and Y. Velner. Hyperplane separation technique for multidimensional mean-payoff games. In *CONCUR*, 2013.
10. A. Degorre, L. Doyen, R. Gentilini, J.-F. Raskin, and S. Torunczyk. Energy and mean-payoff games with imperfect information. In *CSL*, 2010.
11. T. Tomita, S. Hiura, S. Hagihara, and N. Yonezaki. A temporal logic with mean-payoff constraints. In *ICFEM*, 2012.
12. Y. Velner. The complexity of mean-payoff automaton expression. In *ICALP (2)*, 2012.
13. Y. Velner. Finite-memory strategy synthesis for robust multidimensional mean-payoff objectives. In *CSL-LICS*, 2014.
14. Y. Velner, K. Chatterjee, L. Doyen, T. A. Henzinger, A. Rabinovich, and J.-F. Raskin. The complexity of multi-mean-payoff and multi-energy games. *CoRR*, 2012.
15. Y. Velner and A. Rabinovich. Church synthesis problem for noisy input. In *FOSACS*, 2011.